(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 November 2002 (14.11.2002)

PCT

(10) International Publication Number
WO 02/091170 A1

(51) International Patent Classification[7]: G06F 9/44, 17/30

(21) International Application Number: PCT/GB02/01978

(22) International Filing Date: 1 May 2002 (01.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/848,828 4 May 2001 (04.05.2001) US

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; P.O. Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB).

(72) Inventors: **JAMES, Zackary, Antone**; 8608 Wildwood Forest Road, Raleigh, NC 27616 (US). **RAJARAMAN, Bala**; 109 Cromwell Court, Cary, NC 27513 (US).

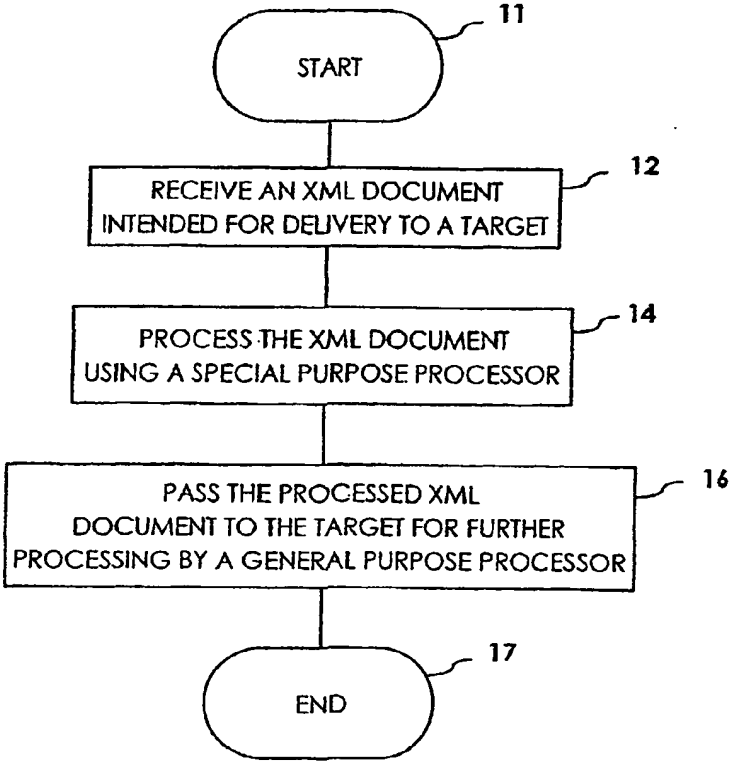
(74) Agent: **BURT, Roger, James**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: DEDICATED PROCESSOR FOR EFFICIENT PROCESSING OF DOCUMENTS ENCODED IN A MARKUP LANGUAGE



(57) Abstract: A dedicated processor for efficient processing of documents encoded in a markup language, such as XML. The dedicated processor is capable of performing traditional parsing, transformation and manipulation processes on the document. The special purpose processor frees a general purpose processor to perform other tasks, resulting in an increase in system performance. In one embodiment, the dedicated processor includes a general purpose processor and suitable software which is provided in addition to the general purpose processor which has been traditionally used for processing. In such an embodiment, the dedicated processor may be implemented in a multi-processor system. In another embodiment, the dedicated processor is implemented in special purpose hardware, e.g. as an integrated circuit embodied in silicon in one or more chips. In either embodiment, the dedicated processor may be provided to offload processing locally or remotely.

WO 02/091170 A1

European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

DEDICATED PROCESSOR FOR EFFICIENT PROCESSING OF DOCUMENTS ENCODED IN A MARKUP LANGUAGE

FIELD OF THE INVENTION

The present invention relates generally to documents encoded in a markup language, such as eXtensible Markup Language (XML), and particularly to processing of XML documents in XML environments, such as a communications network.

DESCRIPTION OF THE RELATED ART

Business and consumer use of distributed computing, e.g network computing, has gained tremendous popularity in recent years. For business purposes, there are two main categories of network interactions between computing elements of distributed computing, namely, those that connect users to business processes and those that interconnect the business process elements. An example of the first is the traditional Web whereby a user may use Web browser software to interact with business data and applications at a Web server using the HyperText Markup Language (HTML) data format transported by the HyperText Transport Protocol (HTTP). An example of the second is traditional "Electronic Document Interchange" (EDI) whereby documents such as requisitions, purchase orders, invoice, shipping notification, etc. existing in standardized electronic formats (such as ANSI X.12 or UN/EDIFACT) are moved between organizational processes by protocols such as X.400, SNADS, TMR, SMTP, etc. For both categories of network interactions, there is a trend toward using the HTTP Web transport protocol and a common data format known as eXtensible Markup Language ("XML").

XML is a tag language, which is a language that uses specially-designated constructs referred to as "tags" to delimit (or "mark up") information. In the general case, a tag is a keyword that identifies the data that is associated with the tag, and is typically composed of a character string enclosed in special characters, i.e., letters and numbers which are defined and reserved for use with tags so that a parser processing the data stream will recognize the tag.

The popularity of XML is due in part to its extensible and flexible syntax, which allows document developers to create tags to convey an explicit nested tree document structure (where the structure is determined

from the relationship among the tags in a particular document). Document developers can define their own tags which may have application-specific semantics. Because of this extensibility, XML documents may be used to specify many different types of information, for use in a virtually unlimited number of contexts. A number of XML derivative notations have been defined, and continue to be defined, for particular purposes. "VoiceXML" is an example of one such derivative. References herein to "XML" are intended to include XML derivatives and semantically similar notations such as derivatives of the Standard Generalized Markup Language, or "SGML", from which XML was derived. Refer to ISO 8879, "Standard Generalized Markup Language (SGML)", (1986) for more information on SGML. Refer to "Extensible Markup Language (XML), W3C Recommendation 10-February-1998" which is available on the World Wide Web at http://www.w3.org/TR/1998/REC-xml-19980210, for more information on XML.

The extensible tag syntax enables an XML document to be easily human-readable, e.g. to convey the semantic meaning of the associated data values and the overall relationship among the elements of the data. This human-friendly, well-structured format enables a human being to quickly look through an arbitrary XML document and understand the data and its meaning. However, the raw content of most XML documents will never be seen by a human: instead, what the end user sees is typically created using a rendering application (such as an XML parser within a browser) which strips out the tags and displays only the embedded data content. The added overhead of the human-friendly tag syntax makes processing, e.g. parsing, of the document burdensome to the processor. Typically, an XML document is parsed and stored internally as a Document Object Model (DOM) tree representation by an XML parser. DOM trees are physically stored in a tree representation, using objects to represent the nodes in the tree, the attributes of the nodes, the values of the nodes, etc.

Transformations, i.e. operations, are then performed (e.g. by content renderers or style sheet processors) by operating upon this tree representation. For example, a particular transformation may include deleting elements from a document by pruning subtrees from the DOM tree; or renaming elements within a document by traversing the DOM tree to find the occurrences of the element name, and substituting the new name into the appropriate nodes of the DOM tree. (DOM is published as a Recommendation of the World Wide Web Consortium ("W3C"), titled "Document Object Model (DOM) Level 1 Specification, Version 1.0" (1998) and available on the Web at http://www.w3.org/TR/REC-DOM-Level-1. "DOM" is a trademark of Massachusetts Institute of Technology.) The type of

transformation is typically target dependent. For example, such transformation may be performed according to an intended recipient's registered preferences or according to capabilities of a target device, e.g. a Web-enabled wireless telephone. Transformations are very processor intensive and are becoming more prevalent, and thus more burdensome, as a broader range of heterogeneous devices seek to access a common set of data.

The parsing, including creation of a DOM tree, and transformation of documents is typically performed by special purpose software executed by a general purpose hardware processor. For example, these steps are typically performed by a server on an edge of a network, e.g. using a WebSphere® Transcoding Product (WTP) special purpose software manufactured and/or distributed by International Business Machines Corporation of Armonk, New York, U.S.A. ("IBM") and executable by a general purpose processor, such as a standard PC's microprocessor.

In some embodiments, the document tree may be manipulated to create a document array model structure, as is generally known in the art. Generally, in an array model, data is organized to represent an ordered set of values that can be accessed by supplying one or more values which uniquely identify one of the values of the set. Accordingly, human-friendly markup language tags are represented in an array model rather than a tree model. The array model simplifies and expedites processing.

In addition, XML documents can be transformed into or represented in the mXML language, a machine-oriented language similar to XML. The mXML notation is more compact than the human-friendly XML notation and therefore provides performance gains in processing and transmission.

The parsing, transformation and other manipulation steps, e.g. XML document recognition, content based style sheet selection, content based routing and other traditional XML processing steps, are tremendously processor intensive, which is burdensome on the general purpose processor and other system resources. Specifically, such processing steps prevent or delay the general purpose processor from performing other tasks required of the general purpose processor.

SUMMARY OF THE INVENTION

The present invention accordingly provides, in a first aspect, a method for efficient processing of a document encoded in a markup language, the method comprising the steps of: receiving a document intended for delivery to a target; processing the document using a special purpose processor; and passing the processed document to the target for further processing by a general purpose processor.

Preferably, said processing step comprises parsing the document.

Preferably, said processing step comprises performing a transformation on the document.

Preferably, said processing step comprises creating an array- based model of the document.

Preferably, said processing step comprises creating a tree-based model of the document.

Preferably, said special purpose processor comprises an integrated circuit configured for parsing the document.

Preferably, said special purpose processor comprises a supplemental general purpose processor for executing computer readable code for parsing the document, said supplemental general purpose processor being distinct from a primary general purpose processor.

Preferably, said passing step comprises communicating the document, as processed, to an application process through a bus of a printed circuit board.

Preferably, said passing step comprises communicating the document, as processed, to a target via a communications network .

Preferably, the target is a local application process.

Preferably, the target is a remote device.

In a second aspect, the present invention provides a system for efficient processing of a document encoded in a markup language, the system comprising: a memory; a general purpose processor operatively

connected to said memory for executing computer readable code stored in said memory; and a special purpose processor operatively connected to said memory for processing documents encoded in the markup language; wherein said special purpose processor is a dedicated processor.

Preferably, said special purpose processor is configured for parsing documents encoded in machine-oriented extensible markup language (mXML).

Preferably, said special purpose processor is configured for transforming documents encoded in machine-oriented extensible markup language (mXML).

Preferably, said special purpose processor comprises an integrated circuit configured for processing the document.

The system preferably further comprises a telecommunications device operatively connected to said general purpose processor and capable of communicating via a communications network; and a first program stored in said memory and executable by said general purpose processor for controlling said special purpose processor to process the document, and for communicating the document, as processed, to a target.

The system preferably further comprises a second program stored in the memory and executable by said general purpose processor for recognizing the document as encoded in the markup language and responsively controlling said special purpose processor to process the document.

Preferably, said special purpose processor comprises a supplemental general purpose processor for executing computer readable code for processing the document.

Preferably, said computer readable code is configured for processing the document in machine-oriented extensible markup language (mXML).

The system preferably further comprises: a telecommunications device operatively connected to said general purpose processor and capable of communicating via a communications network; and a first program stored in said memory and executable by said general purpose processor for controlling said special purpose processor to process the document, and for communicating the document, as processed, to a target.

The system preferably further comprises: a second program stored in the memory and executable by said general purpose processor for recognizing the document as encoded in the markup language and responsively controlling said special purpose processor to process the document.

The present invention may suitably be embodied in a printed circuit board comprising: a general purpose processor for executing computer readable code stored in a memory; and a special purpose processor operably connected to said general purpose processor for communicating therewith, said special purpose processor being configured for processing documents encoded in a markup language.

Preferably, said special purpose processor comprises an integrated circuit configured for processing the document.

Preferably, wherein said processing includes parsing and/or transforming of the document.

Preferably, said special purpose processor comprises a supplemental general purpose processor.

The printed circuit board preferably further comprises:

a memory operably connected to said supplemental general purpose processor; and

computer readable code stored in said memory and executable by said supplemental general purpose processor for processing the document. There is thus preferably provided a special purpose, dedicated processor for processing documents encoded in a markup language such as XML which can free the general purpose processor to perform other tasks, and at least a hardware-based dedicated processor which can provide for optimization of processing steps by eliminating or reducing inefficiencies in human-friendly software code of the type heretofore known by relying on machine language characteristics.

The present invention preferably provides a method for efficient processing of a document encoded in a markup language, the method comprising the step of: communicating an array-based data model representing the document to an application process through a bus of a printed circuit board.

Preferably, said data model represents a document encoded in mXML.

Preferably, said data model represents a document encoded in XML.

The present invention preferably provides a method and apparatus for efficient processing of documents using a dedicated (special purpose) processor. The dedicated processor is preferably capable of performing traditional parsing, transformations and manipulation processes, e.g. on an XML document. Conceptually, the use of a special purpose processor for processing the document frees the general purpose processor to perform other tasks, resulting in an increase in system performance. In other words, the dedicated processor preferably does not compete for system resources.

In one embodiment, the dedicated processor is implemented in special purpose hardware, e.g. an integrated circuit embodied in one or more silicon chips. This is particularly advantageous because it allows use of machine code and other speed-related advantages typical of hardware implementations. For example, performance can be improved by configuring the dedicated processor to process mXML documents, by first converting XML documents to mXML if necessary. This is particularly advantageous in a hardware- based embodiment. Configuring the dedicated processor to represent documents in array- based notation can also be used to enhanced performance, e.g. in mXML-based embodiments. A hardware implementation is particularly useful in a single processor computer system, e.g. as a hardwired chip in communication with the general purpose processor.

In another embodiment, the dedicated processor includes a general purpose processor and suitable software which is provided in addition to the general purpose processor which has been traditionally used for processing documents encoded in a markup language. For example, one of several general purpose processors in a multi-processor computer system may be designated as the dedicated processor.

In either embodiment, the dedicated processor may be provided remotely, e.g. in a processing device which receives and processes documents before receipt by the intended target. An arrangement is which the dedicated processor is network accessible has been found particularly advantageous because it is capable of supporting numerous devices and thereby offloading processing for numerous devices. Alternatively, in either a hardware- or software-based embodiment, the dedicated processor

may be provided locally in the target device, e.g. co-located with a general purpose processor in a single device.

The present invention preferably provides a method for efficient processing of a document encoded in a markup language, the method comprising the step of communicating an array- based data model representing the document to an application process through a bus of a printed circuit board. The present invention further preferably provides a method for efficient processing of a document encoded in a markup language comprising the steps of receiving a document intended for delivery to a target, processing the document using a special purpose processor, and passing the processed document to the target for further processing by a general purpose processor.

DESCRIPTION OF THE DRAWINGS

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 provides a flowchart which sets forth an overview of exemplary logic for processing documents in accordance with a preferred embodiment.

Figure 2 provides a flowchart which sets forth a first embodiment of exemplary logic for processing documents in accordance with Figure 1;

Figure 2B provides a flowchart which sets forth a second embodiment of exemplary logic for processing documents in accordance with Figure 1;

Figure 3 is a diagram of a networked computing environment in which the present invention may be practised; and

Figure 4 is a block diagram of a computer workstation environment in accordance with a preferred embodiment.

DETAILED DESCRIPTION

Figure 1 provides a flowchart 10 which sets forth an overview of exemplary logic for processing documents in accordance with the present invention. As used herein, "processing" refers to parsing, transforming, e.g. applying a style sheet and/or adding/modifying/deleting data from a

document/document tree and/or formatting data and other traditional XML processing steps, including XML encoded document recognition, content based routing, etc. The exemplary logic may be used by a hardware-based or software-based implementation of the special purpose processor in accordance with the present invention, as discussed further below.

As shown in Figure 1, the method starts with receipt of a document, e.g. an XML document, intended for delivery to a target, as shown at steps 11 and 12. As used herein, the target could be a target device or a target application process, such as a web browser, business-to-business environment process, business-to-client environment process, business logic process, back-end server process, edge server process, web service information exchange process, etc. The document is then processed using a special purpose processor in accordance with the present invention, as shown at step 14. This relieves a general purpose processor, which has heretofore been used to perform such processing, of the intensive processing which typically significantly burdens system resources. In other words, the processing of the document is offloaded from the general purpose processor which traditionally has performed such processing. The processed document is then passed to the target for further processing, e.g. post-processing including rendering, another transformation, routing to another application process, etc., as shown at step 16. Such post-processing is performed by the general purpose processor, as is well known in the art. It may be advantageous to perform such post-processing at the target because. However, the most- intensive processing has been effectively offloaded to the special purpose processor. This greatly enhances system performance. The method then ends, as shown at step 17.

Accordingly, the special purpose processor receives as input an XML document in the form of DOM, DAM, MXML or STREAM and a style sheet. In addition, a node tree associated with the document is communicated to an application process through a bus of a printed circuit board. This occurs regardless of whether the special purpose processor is hardware or software-based (as discussed further below), or whether the special purpose processor is located locally or remotely, as discussed further below. This communication also results regardless of whether the document is transformed or otherwise manipulated after parsing, or a combination thereof.

The added overhead of the human-friendly tag syntax makes processing, e.g. parsing to create the DOM tree, of the document burdensome to the general purpose processor. This burden is unnecessary

when the documents will only be "seen" by a computer program, such as for those documents which are formatted for interchange between computer programs for business-to-business ("B2B") or business-to-consumer ("B2C") use.

One way to improve processing efficiency is to abandon the human-friendly tag structure. The assignee hereof has previously developed a machine-oriented notation for use as an XML alternative. The machine-oriented notation improves processing time for arbitrarily-structured documents and reduces the storage requirements and transmission costs of data interchange while still retaining the extensibility and flexibility of XML and while conveying equivalent content and semantic information. This machine-oriented notation is referred to herein as "mXML". Accordingly, in a preferred embodiment, the dedicated processor is configured to understand and interpret mXML, thereby resulting in processing efficiencies.

Creation of a DOM tree is computationally expensive in terms of processing time and memory requirements. Using this tree-oriented DOM representation as an internal storage format requires a considerable amount of memory and/or storage space to store the required objects. In addition, a number of computer program instructions must be executed to allocate memory and create the objects, delete objects and de-allocate memory, and traverse the tree structure to perform operations thereon. Execution of these instructions increases the processing time required for structured documents, as do the operating system-invoked instructions which are periodically executed to perform "garbage collection" (whereby the space being used by objects can be reclaimed after the objects have been logically deleted or de-allocated).

Another way to improve processing efficiency is to use an array-based notation. The Xalan XSLT (Extensible Language Transformations) processor from the Apache Software Foundation reduces the number of objects used by DOM processors somewhat by providing an in-memory Document Table Model ("DTM") representation of a DOM tree. An array is used instead of a set of "real objects" for storing the DOM tree itself. However, there are still many objects around to represent the XML data content of a document (including objects for the nodes, node values, attributes, attribute values, etc.). Array-based processing makes it easier to navigate the tree structure, e.g. for transformation purposes, etc. Accordingly, by implementing array-based processing into the dedicated processor, further performance gains are realized. In a highly

preferred embodiment, the dedicated processor is configured to process a document using an array-based notation.

Figure 2A provides a flowchart 20 which sets forth a first embodiment of exemplary logic for processing documents in accordance with Figure 1. In the example of Figure 2A, a hardware-based special purpose processor is provided remotely, e.g. as a special purpose chip or chipset in a network-accessible processing device. Specifically, the special purpose processor is provided at a device different from the device where the general purpose processor that has traditionally performed the software-based processing of such documents resides. For example, this arrangement is advantageous in network-based applications, e.g. by providing a network accessible having a special purpose processor for offloading processing from, and thereby supporting, numerous devices. Alternatively, the special purpose processor may be provided locally, i.e. in the same device where the general purpose processor that has traditionally performed the software-based processing of such documents. For example, the special purpose processor may be provided locally to offload processing from an associated general purpose processor. In other words, when provided locally, the special purpose processor offloads processing from a general purpose processor within the same device. When provided remotely, the special purpose processor offloads processing from a general purpose processor within a remote device. Advantageously, the offloaded processing is conducted in a manner that is transparent to the user.

Figure 3 is a diagram of a networked computing environment in which a remotely provided hardware-based special purpose processor according to the present invention may be practiced. The network of Figure 3 is discussed in greater detail below. For the example of Figure 2A, consider that gateway server 346 of Figure 3 is a processing device having a hardware-based special purpose processor as described above. In this example, device 310a is a personal computer device 310a that is connected to server 346 by a communications network. Consider that device 310a is the target device for an XML document served by data server 348. More specially, consider that web browser software being executed by a general purpose processor within device 310a is the target application process. Typical web browser software is capable of processing HTML, but not XML. Accordingly, a JAVA or other plug- in software application is typically executed by a general purpose processor within the device to translate the XML to HTML for post-processing, e.g. interpretation and display, by the web browser and general purpose processor. This places a burden on the

general purpose processor of the target devices to convert XML to HTML. Accordingly, in this example, server 346 is provided with a hardware-based special purpose processor for processing XML documents. In the example of Figure 2A, and as shown in Figure 3, an XML document deliverable to device 310a from data server 348 is first received (and implicitly recognized as such by a hardware or software based recognition engine) at an intermediate processing device (server 346) as shown at step 22 of Figure 2A. The XML document is then processed, e.g. parsed by the hardware-based special processor of server 346, as shown at step 24 of Figure 2A. For example, such parsing results in creation of a document tree data model representing the XML document, e.g. in document object model (DOM) format. Alternatively, the special purpose processor of device 346 is configured parse the document to create a data model in document array model (DAM) format.

Optionally, e.g. if required for the target device, the document is further processed to perform a transformation, as shown at step 26 of Figure 2A. For example, such transformations are typically performed to format content deliverable to handheld devices such as personal digital assistant (PDA) device 310b or web-enabled wireless telephone 310c of Figure 3. For example, such transformations are now typically performed by IBM's Websphere® Transcoding Product (WTP) software, e.g. stored on gateway server 346 of Figure 3. Using the special purpose processor to perform such transformation provides a substantial improvement in system performance (e.g. in processing device 346). The particular transformation required is typically device specific, e.g. to provide lower-resolution or no images, etc., or user-specific, e.g. according to a user-preference profile, for example, to eliminate certain types of content.

Referring again to Figure 2A, the processed, e.g. parsed and/or transformed, XML document is transmitted via a communications network to the target device for post- processing by the target device's general purpose processor, as shown at step 28. For example, this step may be performed by the CPU of personal computer device 310b of Figure 3, e.g. to display the document via web browser software. The process then ends, as shown at step 29. In this manner, burdens on the general purpose processor of the target device normally associated with parsing and/or transforming of the document are eliminated by offloading such burdens to the special purpose processor of the processing device, e.g. server 346.

Figure 2B provides a flowchart 30 which sets forth a second embodiment of exemplary logic for processing documents in accordance with Figure 1. In the example of Figure 2B, a software-based special purpose processor is provided. Although it is possible to provide the software-based processor remotely, in this example, the software-based processor is provided locally, i.e. to offload processing from a general purpose processor within the same device. For example, this arrangement is advantageous in multi-processor systems, and in systems which are not capable of communicating via a communications network.

The networked computing environment of Figure 3 can also be used to practice the invention according to the logic set forth in Figure 2B. For the example of Figure 2B, consider that server 346 of Figure 3 is a multi-processor processing device and that a software-based special purpose processor is implemented in server 346 by dedicating one of the general purpose processors to the task of XML processing, by running software stored in the memory of server 346. For this example, consider that server 346 is the target, unlike the example of Figure 2A discussed above in which the workstation 310a was the target and processing was being offloaded from the workstation 310a to the special processor of server 346 (a remotely provided special purpose processor example). In this example, processing is being offloaded from the general purpose processor of server 346 to the special purpose processor of server 346. A processing device in accordance with the present invention is discussed in detail below with reference to Figure 4.

Referring to Figures 2B and 3, the process starts with receipt of an XML document at the processing (in this case target) device, as shown at steps 31, 32 of Figure 2B. The XML document is then parsed and transformed by the special purpose processor, as shown at steps 34 and 36 of Figure 2B. These steps are similar to steps 24 and 26 of Figure 2A. However, in this example, these steps are performed by the local special purpose processor 432 (in this example a general purpose processor which runs software stored in the memory 418, 430 of the workstation 410) of the processing device of Figure 4. The parsed and/or transformed XML document is then passed to the general purpose processor, as shown at step 38, e.g. for post-processing. For example, this step includes communicating a node tree representing the document to an application process running locally through a bus of a printed circuit board. Because the special processor is provided locally, this step need not include transmitting the processed document via a communications network, as in the example of Figure 2A.

In summary, a special processor could be provided locally in server 346 of Figure 3 to offload processing from a general purpose processor in server 346 (a local embodiment) or to offload processing from a workstation, e.g. 310b (a remote embodiment). In either of the local or remote embodiments, the special processor may be provided in either a hardware implementation (a special purpose chip or chipset) or a software implementation (an additional general purpose processor and special purpose software).

NETWORK

Figure 3 illustrates an exemplary data processing network 340 in which the present invention may be practiced. The data processing network 340 may include a plurality of individual networks, such as wireless network 342 and network 344, each of which may include a plurality of individual workstations/devices, e.g. 410a, 410b, 410c. Additionally, as those skilled in the art will appreciate, one or more LANs may be included (not shown), where a LAN may comprise a plurality of intelligent workstations coupled to a host processor.

The networks 342 and 344 may also include mainframe computers or servers, such as a gateway computer 346 or application server 347 (which may access a data repository 348). A gateway computer 346 serves as a point of entry into each network 344. The gateway computer 346 may be preferably coupled to another network 342 by means of a communications link 350a. The gateway computer 346 may also be directly coupled to one or more workstations, e.g 310d, 310e using a communications link 350b, 350c. The gateway computer 346 may be implemented using any appropriate processor, such as IBM's Network Processor. For example, the gateway computer 346 may be implemented using an IBM pSeries (RS/6000) or xSeries (Netfinity) computer system, an Enterprise Systems Architecture/370 available from IBM, an Enterprise Systems Architecture/390 computer, etc. Depending on the application, a midrange computer, such as an Application System/400 (also known as an AS/400) may be employed. ("Enterprise Systems Architecture/370" is a trademark of IBM; "Enterprise Systems Architecture/39", "Application System/400", and "AS/400" are registered trademarks of IBM.) These are merely representative types of computers with which the present preferred embodiment may be used.

The gateway computer 346 may also be coupled 349 to a storage device (such as data repository 348). Further, the gateway 346 may be directly

or indirectly coupled to one or more workstations/devices 310d, 310e, and servers such as application server 347.

Those skilled in the art will appreciate that the gateway computer 346 may be located a great geographic distance from the network 342, and similarly, the workstations/devices may be located a substantial distance from the networks 342 and 344. For example, the network 342 may be located in California, while the gateway 346 may be located in Texas, and one or more of the workstations/devices 310 may be located in New York. The workstations/devices 310 may connect to the wireless network 342 using a networking protocol such as the Transmission Control Protocol/Internet Protocol ("TCP/IP") over a number of alternative connection media, such as cellular phone, radio frequency networks, satellite networks, etc. The wireless network 342 preferably connects to the gateway 346 using a network connection 350a such as TCP or UDP (User Datagram Protocol) over IP, X.25, Frame Relay, ISDN (Integrated Services Digital Network), PSTN (Public Switched Telephone Network), etc. The workstations/devices 310 may alternatively connect directly to the gateway 346 using dial connections 350b or 350c. Further, the wireless network 342 and network 344 may connect to one or more other networks (not shown), in an analogous manner to that depicted in Figure 3.

The present preferred embodiment may be used on a client computer or server in a networking environment, or on a standalone workstation (for example, to prepare a file or to process a file which has been received over a network connection, via a removable storage medium, etc.). (Note that references herein to client and server devices are for purposes of illustration and not of limitation: the present preferred embodiment may also be used advantageously with other networking models.) When used in a networking environment, the client and server devices may be connected using a "wireline" connection or a "wireless" connection. Wireline connections are those that use physical media such as cables and telephone lines, whereas wireless connections use media such as satellite links, radio frequency waves, and infrared waves. Many connection techniques can be used with these various media, such as: using the computer's modem to establish a connection over a telephone line; using a LAN card such as Token Ring or Ethernet; using a cellular modem to establish a wireless connection; etc. The workstation or client computer may be any type of computer processor, including laptop, handheld or mobile computers; vehicle-mounted devices; desktop computers; mainframe computers; etc., having processing (and, optionally, communication) capabilities. The server, similarly, can be one of any number of different types of computer

which have processing and communication capabilities. These techniques are well known in the art, and the hardware devices and software which enable their use are readily available.

PROCESSING DEVICE

Figure 4 is a block diagram of a processing device 410 in accordance with the present preferred embodiment. The exemplary processing device 410 is representative of workstation 310a or server 346 of Figure 3, as discussed above. This block diagram represents hardware for a local implementation or a remote implementation. However, appropriate software is provided, e.g. stored in the memory, to configure the workstation to offload processing from a local and/or a remote general purpose processor.

As is well known in the art, the workstation of Figure 4 includes a representative processing device, e.g. a single user computer workstation 410, such as a personal computer, including related peripheral devices. The workstation 410 includes a general purpose microprocessor 412 and a bus 414 employed to connect and enable communication between the microprocessor 412 and the components of the workstation 410 in accordance with known techniques. The workstation 410 typically includes a user interface adapter 416, which connects the microprocessor 412 via the bus 414 to one or more interface devices, such as a keyboard 418, mouse 420, and/or other interface devices 422, which can be any user interface device, such as a touch sensitive screen, digitized entry pad, etc. The bus 414 also connects a display device 424, such as an LCD screen or monitor, to the microprocessor 412 via a display adapter 426. The bus 414 also connects the microprocessor 412 to memory 428 and long-term storage 430 (collectively, "memory") which can include a hard drive, diskette drive, tape drive, etc.

The workstation 410 may communicate with other computers or networks of computers, for example via a communications channel or modem 434. Alternatively, the workstation 410 may communicate using a wireless interface at 434, such as a CDPD (cellular digital packet data) card. The workstation 410 may be associated with such other computers in a LAN or a wide area network (WAN), or the workstation 410 can be a client in a client/server arrangement with another computer, etc. All of these configurations, as well as the appropriate communications hardware and software, are known in the art.

In accordance with the present preferred embodiment, a special purpose processor 432 is provided in communication with general purpose microprocessor 412, memory 428, long term storage device 430, etc. by bus 414. When used to offload processing from a local general purpose processor, the workstation 410 provides exceptional performance improvements because of the proximity and/or priority of the special processor to the general purpose processor from which processing tasks are offloaded.

In the software-based example of Figure 2B, the special purpose processor 432 includes a dedicated general purpose microprocessor running processing software stored in the memory 428 and/or storage device 430. In a hardware-based embodiment, the special purpose processor 432 includes a special purpose chip or chipset. In either embodiment, additional performance gains can be realized by configuring the special purpose processor to use array-based processing and/or machine language based processing, e.g. mXML. Additional performance gains can be realized by optimizing the hardware-based embodiment to use such array-based processing and/or mMXL. For example, the special purpose processor 432 may be implemented through a combination of special purpose hardware and microcode that may also include a general purpose processor that offloads nonrepetitive tasks from the special purpose processor, e.g. to handle infrequent software functions such as processing style sheet updates, managing personalization or content/data, caching, etc.

CLAIMS

1. A method for efficient processing of a document encoded in a markup language, the method comprising the steps of:

receiving a document intended for delivery to a target;

processing the document using a special purpose processor; and

passing the processed document to the target for further processing by a general purpose processor.

2. A method as claimed in claim 1, wherein said processing step comprises parsing the document, or wherein said processing step comprises performing a transformation on the document.

3. A method as claimed in claim 1 or claim 2, wherein said processing step comprises creating an array- based model of the document, or wherein said processing step comprises creating a tree-based model of the document.

4. A method as claimed in any preceding claim, wherein said special purpose processor comprises an integrated circuit configured for parsing the document, or wherein said special purpose processor comprises a supplemental general purpose processor for executing computer readable code for parsing the document, said supplemental general purpose processor being distinct from a primary general purpose processor.

5. A method as claimed in claim 4, wherein said passing step comprises communicating the document, as processed, to an application process through a bus of a printed circuit board, or wherein said passing step comprises communicating the document, as processed, to a target via a communications network

6. A system for efficient processing of a document encoded in a markup language, the system comprising:

a memory;

a general purpose processor operatively connected to said memory for executing computer readable code stored in said memory; and

a special purpose processor operatively connected to said memory for processing documents encoded in the markup language;
wherein said special purpose processor is a dedicated processor.

7. A system as claimed in claim 6, wherein said special purpose processor is configured for parsing documents encoded in machine-oriented extensible markup language (mXML, or wherein said special purpose processor is configured for transforming documents encoded in machine-oriented extensible markup language (mXML).

8. A system as claimed in claim 6 or claim 7, wherein said special purpose processor comprises an integrated circuit configured for processing the document.

9. A system as claimed in any of claims 6 to 8, further comprising:

a telecommunications device operatively connected to said general purpose processor and capable of communicating via a communications network; and

a first program stored in said memory and executable by said general purpose processor for controlling said special purpose processor to process the document, and for communicating the document, as processed, to a target.

10. A system as claimed in claim 9, further comprising:

a second program stored in the memory and executable by said general purpose processor for recognizing the document as encoded in the markup language and responsively controlling said special purpose processor to process the document.

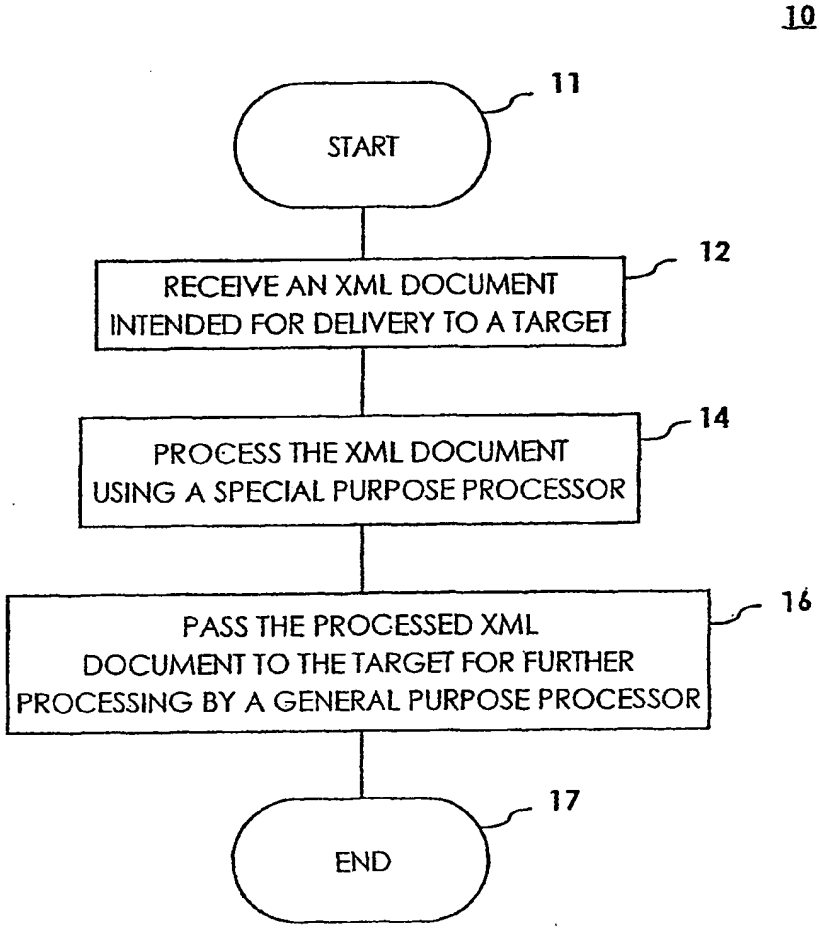

10
11
START
12
RECEIVE AN XML DOCUMENT INTENDED FOR DELIVERY TO A TARGET
14
PROCESS THE XML DOCUMENT USING A SPECIAL PURPOSE PROCESSOR
16
PASS THE PROCESSED XML DOCUMENT TO THE TARGET FOR FURTHER PROCESSING BY A GENERAL PURPOSE PROCESSOR
17
END

Figure 1

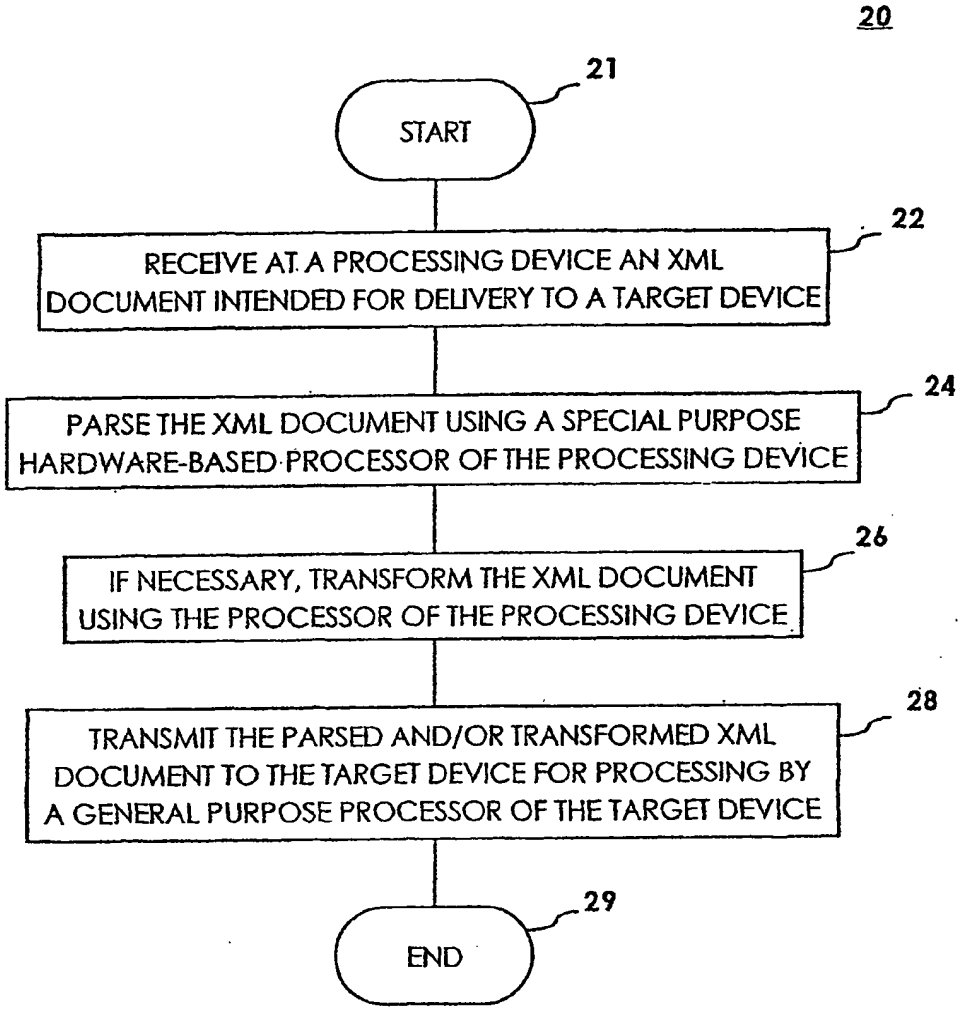

20
21
START
22
RECEIVE AT A PROCESSING DEVICE AN XML DOCUMENT INTENDED FOR DELIVERY TO A TARGET DEVICE
24
PARSE THE XML DOCUMENT USING A SPECIAL PURPOSE HARDWARE-BASED PROCESSOR OF THE PROCESSING DEVICE
26
IF NECESSARY, TRANSFORM THE XML DOCUMENT USING THE PROCESSOR OF THE PROCESSING DEVICE
28
TRANSMIT THE PARSED AND/OR TRANSFORMED XML DOCUMENT TO THE TARGET DEVICE FOR PROCESSING BY A GENERAL PURPOSE PROCESSOR OF THE TARGET DEVICE
29
END

Figure 2A

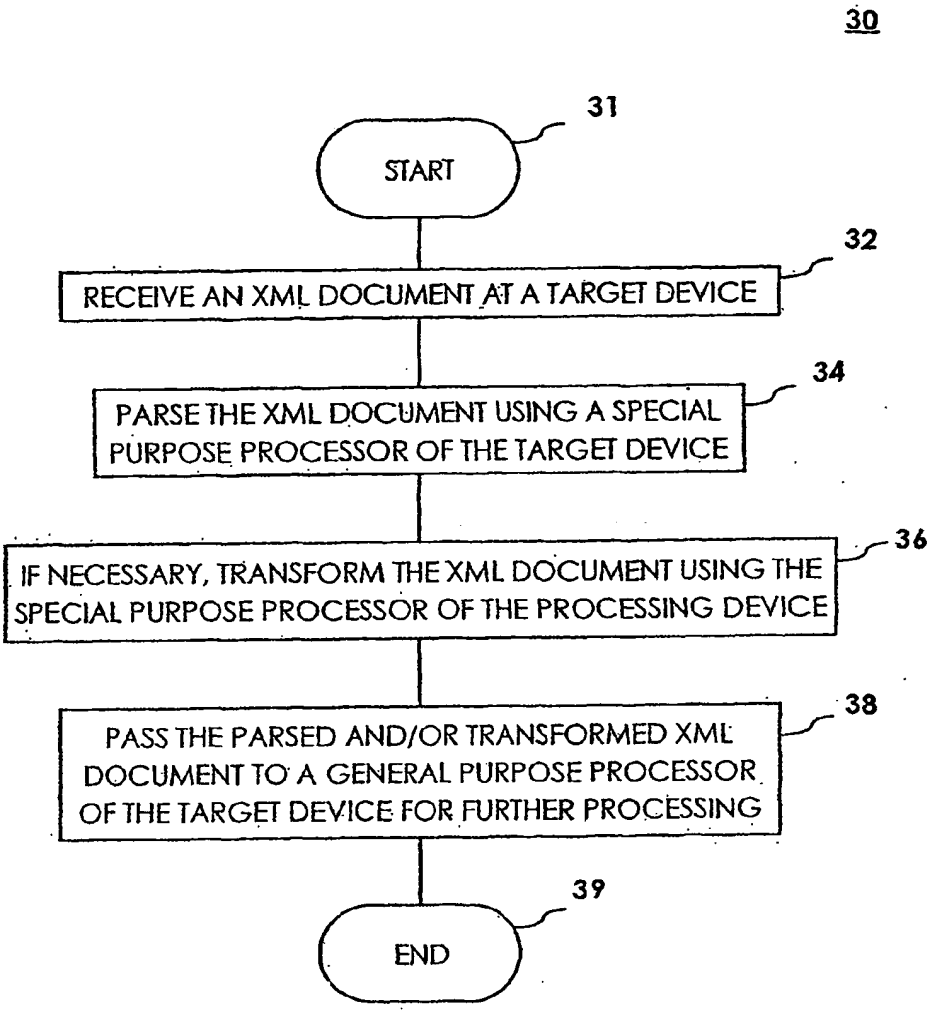

30
31
START
32
RECEIVE AN XML DOCUMENT AT A TARGET DEVICE
34
PARSE THE XML DOCUMENT USING A SPECIAL PURPOSE PROCESSOR OF THE TARGET DEVICE
36
IF NECESSARY, TRANSFORM THE XML DOCUMENT USING THE SPECIAL PURPOSE PROCESSOR OF THE PROCESSING DEVICE
38
PASS THE PARSED AND/OR TRANSFORMED XML DOCUMENT TO A GENERAL PURPOSE PROCESSOR OF THE TARGET DEVICE FOR FURTHER PROCESSING
39
END

Figure 2B

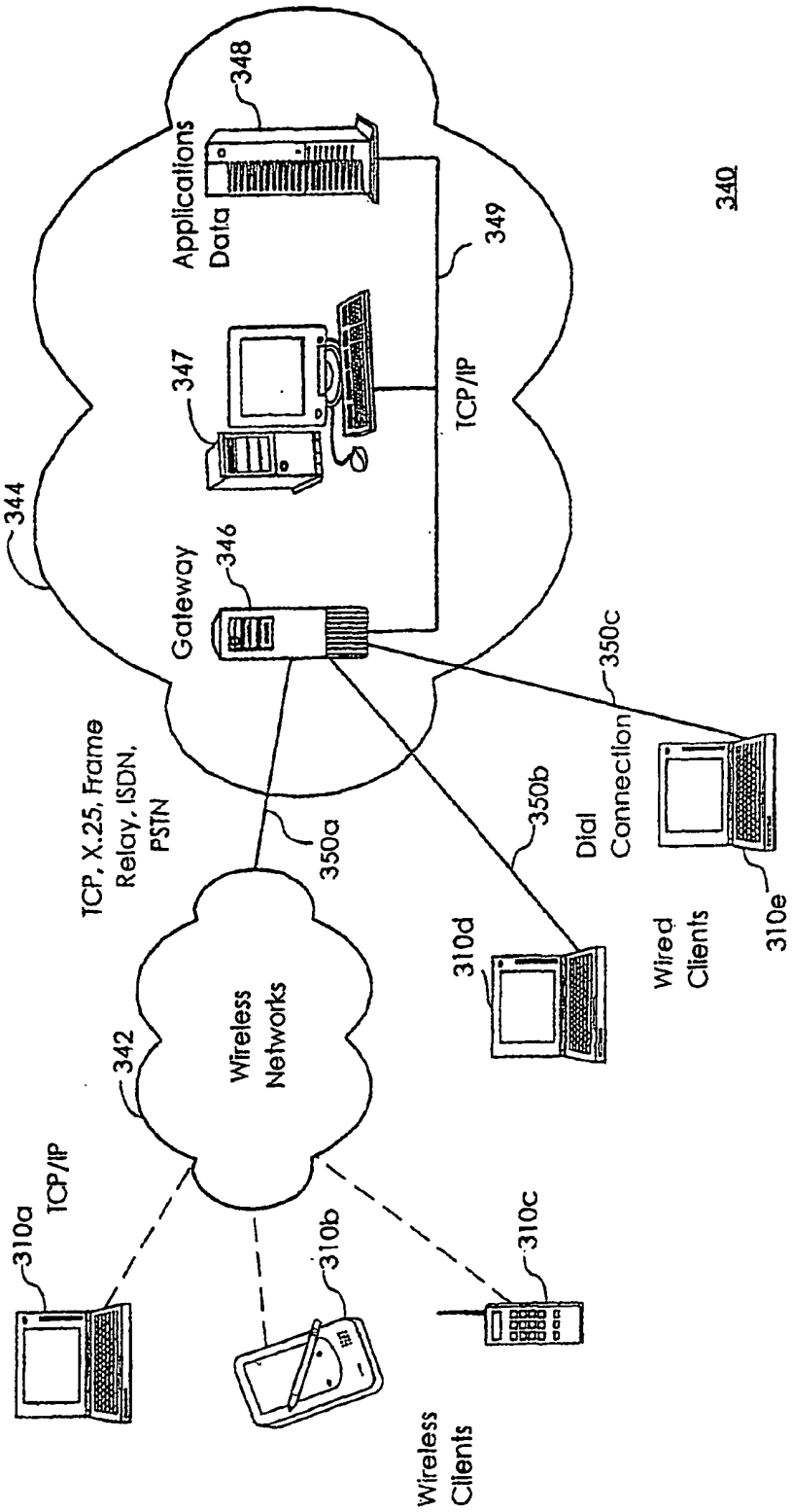

340
344
348
Applications
Data
349
347
TCP/IP
346
Gateway
350c
350b
Dial
Connection
350a
TCP, X.25, Frame
Relay, ISDN,
PSTN
342
Wireless
Networks
310d
Wired
Clients
310e
310a
TCP/IP
310b
310c
Wireless
Clients

Figure 3

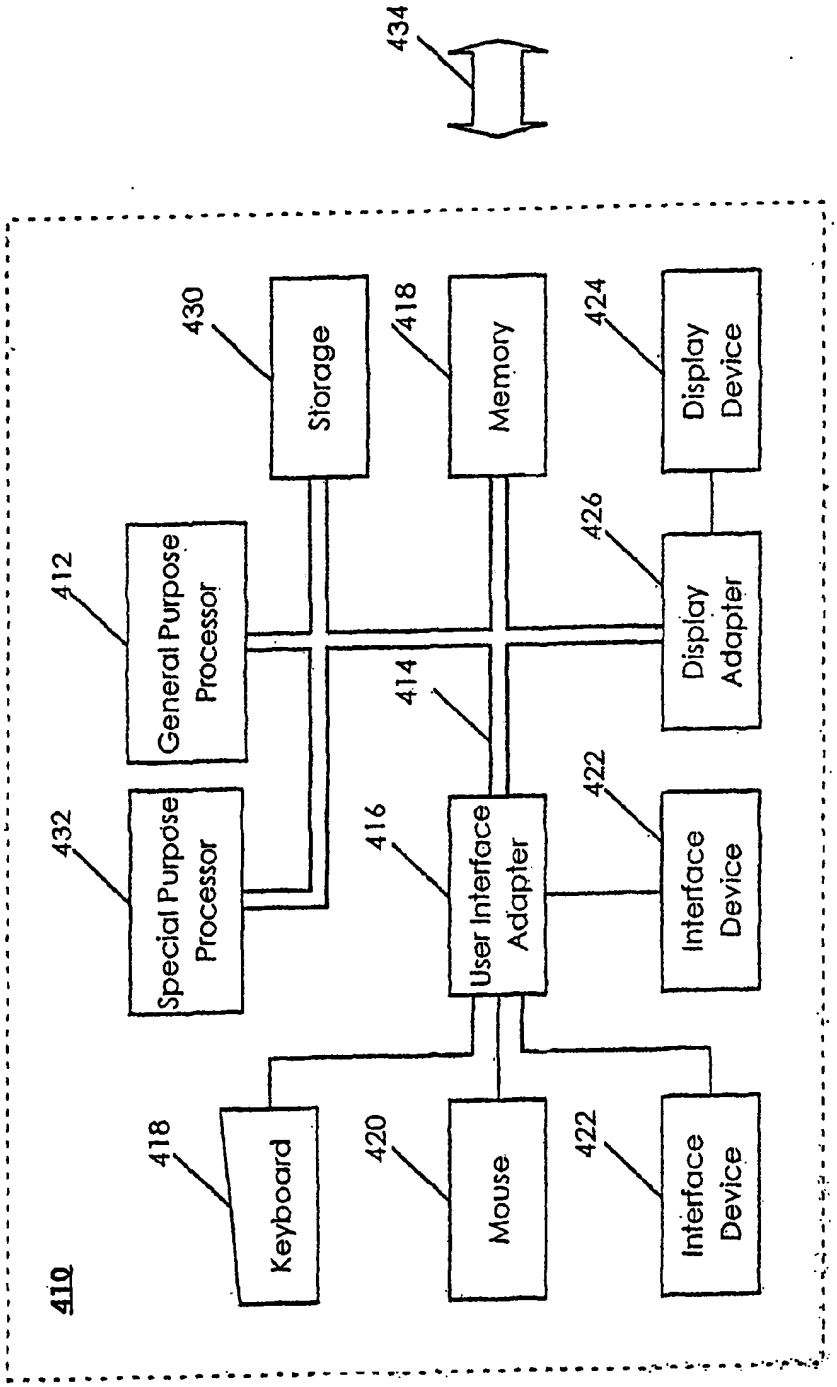

434
410
Keyboard
418
Mouse
420
Interface Device
422
Special Purpose Processor
432
General Purpose Processor
412
User Interface Adapter
416
Interface Device
422
Storage
430
Memory
418
Display Adapter
426
Display Device
424
414

Figure 4

INTERNATIONAL SEARCH REPORT

PCT/GB 02/01978

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/44 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC, IBM-TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00 52564 A (AMULET TECHNOLOGIES LLC) 8 September 2000 (2000-09-08) abstract figure 2 figure 3 page 4, line 19 -page 9, line 11	1-10
P,X	GB 2 357 348 A (IBM) 20 June 2001 (2001-06-20) abstract figure 1 figure 3 figure 4 claim 1 claim 2	1-10

[ ] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 August 2002	05/09/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Sadoune, M-M

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

PCT/GB 02/01978

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 0052564	A	08-09-2000	EP	1090344 A2	11-04-2001
			WO	0052564 A2	08-09-2000
GB 2357348	A	20-06-2001	NONE		